Single Image Deraining Using Time-Lapse Data

Jaehoon Cho, *Student Member, IEEE*, Seungryong Kim, *Member, IEEE*, Dongbo Min, *Senior Member, IEEE*, and Kwanghoon Sohn, *Senior Member, IEEE*

***Abstract*—Leveraging on recent advances in deep convolutional neural networks (CNNs), single image deraining has been studied as a learning task, achieving an outstanding performance over traditional hand-designed approaches. Current CNNs based deraining approaches adopt the supervised learning framework that uses a massive training data generated with synthetic rain streaks, having a limited generalization ability on real rainy images. To address this problem, we propose a novel learning framework for single image deraining that leverages time-lapse sequences instead of the synthetic image pairs. The deraining networks are trained using the time-lapse sequences in which both camera and scenes are static except for time-varying rain streaks. Specifically, we formulate a background consistency loss such that the deraining networks consistently generate the same derained images from the time-lapse sequences. We additionally introduce two loss functions, the structure similarity loss that encourages the derained image to be similar with an input rainy image and the directional gradient loss using the assumption that the estimated rain streaks are likely to be sparse and have dominant directions. To consider various rain conditions, we leverage a dynamic fusion module that effectively fuses multi-scale features. We also build a novel large-scale time-lapse dataset providing real world rainy images containing various rain conditions. Experiments demonstrate that the proposed method outperforms state-of-the-art techniques on synthetic and real rainy images both qualitatively and quantitatively. On the high-level vision tasks under severe rainy conditions, it has been shown that the proposed method can be utilized as a pre-preprocessing step for subsequent tasks.**

***Index Terms*—Single image deraining, convolutional neural networks (CNNs), time-lapse dataset, dynamic fusion module.**

I. Introduction

An IMAGE captured in an outdoor environment frequently suffers from visibility degradation due to various weather conditions such as rain [1]–[7], haze [8], [9], or snow [2], [10]. Especially, on rainy days, outdoor images are degraded by rain streaks that cause undesired artifacts such as intensity fluctuation and occlusion [1]–[7], [11]. In this context, a single image deraining technique serves as an essential pre-processing for various image processing tasks [12]–[14].

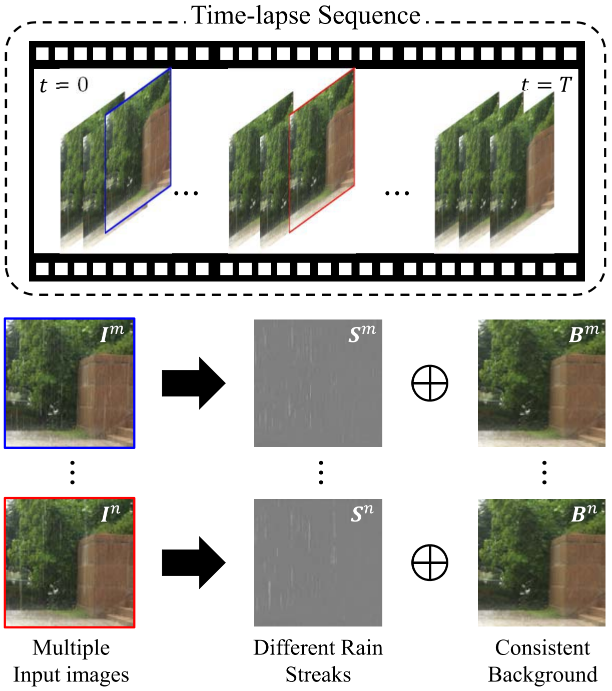


Fig. 1. Illustration of our learning framework for single image deraining: using samples from time-lapse sequences taken at a static scene, we generate corresponding rain streaks and consistent background images for multiple input images from the time-lapse sequences.

Approaches for single image deraining have been traditionally formulated by modeling the physical characteristics of a rain streak, but these hand-designed priors frequently fail to model real rain streaks [6], [7], [15]–[19]. Recently, deep convolutional neural networks (CNNs) based approaches [3]–[5], [20]–[22], [24], [25] have improved the performance of single image deraining substantially. They trained single image deraining networks with a tremendous number of ground truth paired training data in a supervised manner. As constructing such massive training data with real images is extremely difficult, they mostly rely on the synthetic data generated by some rendering tools, such as Photoshop [27] and photo realistic rendering techniques [28]. However, they cannot reflect real environments well, which incurs the domain adaptation issue [5], [22].

To overcome this limitation, we propose a novel learning framework that leverages time-lapse sequences, without using any ground truth paired data. Our approach builds upon the insight that the time-lapse sequences taken under a static

Manuscript received December 23, 2019; revised May 3, 2020; accepted May 25, 2020. Date of publication June 12, 2020; date of current version July 13, 2020. This work was supported in part by the Research and Development Program for Advanced Integrated-Intelligence for Identification (AIID) through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under Grant NRF-2018M3E3A1057289 and in part by the Ewha Womans University Research Grant of 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Emanuele Salerno. *(Corresponding authors: Dongbo Min; Kwanghoon Sohn.)*

Jaehoon Cho and Kwanghoon Sohn are with the School of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, South Korea (e-mail: rehoon@yonsei.ac.kr; khsohn@yonsei.ac.kr).

Seungryong Kim is with the École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland, and also with the Department of Computer Science and Engineering, Korea University, Seoul 02841, South Korea (e-mail: seungryong_kim@korea.ac.kr).

Dongbo Min is with the Department of Computer Science and Engineering, Ewha Womans University, Seoul 03-760, South Korea (e-mail: dbmin@ewha.ac.kr).

Digital Object Identifier 10.1109/TIP.2020.3000612

background with time-varying rain streaks allow to leverage a background consistency constraint, as illustrated in Fig. 1. Concretely, we present a background consistency loss to estimate the consistent backgrounds of input images sampled from time-lapse data. We further propose two additional losses including a structure similarity loss and a directional gradient loss. The first loss encourages the estimated backgrounds to be close to input images, and the latter one enforces the estimated rain streaks to be sparse and have dominant orientations. During training, two images sampled from the time-lapse sequence are fed into the deraining networks. To fully encode various rain streaks, we extract features from multi-scale encoder networks and fuse them using a dynamic fusion module that learns an optimal fusion weight conditionally determined by the input features.

We further construct a novel time-lapse dataset for deraining. Unlike the existing synthetic datasets [3], [5], [20], [22], we collect time-lapse sequences in the real world that contain various rain conditions. To the best of our knowledge, there is no study that exploits the time-lapse sequences of real rainy scenes to learn the single image deraining. Extensive experiments demonstrate that our approach provides state-of-the-art performances on synthetic dataset [3], [5], [22] and even generalizes well on real rainy images [24].

Our main contributions are highlighted as follows.

- We propose to exploit the time-lapse sequences to train the deraining networks, without using any ground-truth paired data.
- To train the networks using the time-lapse sequences, we introduce novel loss functions for enforcing backgrounds to be consistent, and estimated rain streaks to be sparse and have dominant orientations.
- We introduce a large-scale rainy time-lapse dataset.

The remainder of this paper is organized as follows. Section II describes the related works. The proposed method and our time-lapse dataset are presented in Section III. Extensive performance validation is then provided in Section IV, including ablation study and comparison to the state-of-the-arts. Section V concludes this paper.

II. Related Work

A. Single Image Deraining

Classical approaches for single image deraining have used a prior knowledge for modeling a background image or a rain streaks using, e.g., Gaussian mixture model (GMM [18], [19], sparse coding [6], [16], and low-rank constraint [37], [38]. Such hand-crafted methods have shown several limitations such as over-smoothed image details [38] and imperfect separation of rain streaks [16], [34], and frequently failed to model real rainy characteristics.

Recently, deep convolutional neural networks (CNNs) have achieved great success in single image deraining. Fu *et al.* [3] first proposed to solve the deraining task through CNNs. They decomposed rain images into low- and high-frequency parts, and then trained only the high-frequency parts using shallow networks. Yang *et al.* [5] introduced a deep recurrent network to jointly detect and remove rain streaks (JORDER) by designing a multi-task learning architecture using background, rain streak, and binary map indicating a spatial position of rain streaks. Zhang *et al.* [20] adopted the generative adversarial network (GAN) with the perceptual loss to achieve better visual quality of a derained image. Li *et al.* [19] particularly concerned with various rain conditions including sizes and directions of rain streaks. They proposed a multi-stage CNN that consists of several parallel sub-networks to aware different scales of rain streaks. Zhang and Patel [22] proposed a density-aware single image deraining. They designed a multi-stream dense network to characterize a non-uniform rain density. Li *et al.* [39] proposed a recurrent squeeze-and-excitation (SE) based context aggregation network (CAN). The SE block assigned different alpha-values for various rain streaks according to the intensity and transparency, and CAN acquired large receptive field. Li *et al.* [21] proposed a non-locally enhanced encoder-decoder network to efficiently learn increasingly abstract feature representation for rain streaks. Ren *et al.* [40] proposed a simple and progressive recurrent deraining network (PReNet) by repeating a shallow ResNet [41]. Wang *et al.* [24] proposed a spatial attentive network (SPANet) to remove rain streaks in a local-to-global manner. Wei *et al.* [25] proposed a semi-supervised learning approach for single image deraining. Yang *et al.* [42] proposed an extended version of JORDER [5] by exploiting a RNN and a contextualized dilated network for better deraining performance. Yang *et al.* [43] proposed a scale-free network investigating on the scale variety of rain streaks by unrolling a wavelet transform using a recurrent neural network. Fu *et al.* [44] proposed a light-weighted pyramid of network (PyramidDerain) by introducing the mature Gaussian-Laplacian image pyramid decomposition method. However, all these approaches adopt a supervised learning framework using large quantities of paired synthetic training data, which limits their generality and practical use on real world rain images.

B. Video Deraining

Multiple image deraining has also been widely explored. Garg and Nayar [28], [45]–[47] first attempted for rain removal from multiple images. They proposed an appearance model to describe rain streaks, and exploited it to detect rain pixels in videos. Zhang *et al.* [48] focused on investigating the brightness property of rain streaks in videos. Barnum *et al.* [49] proposed a spatio-temporal frequency based method for globally detecting rain streaks using a physical and statistical model. Kim *et al.* [2] proposed to remove rain streaks using temporal correlation with low-rank matrix completion. They subtracted temporally warped frames from the current frame to obtain an initial rain map, and decomposed it into two types of basis vectors using a support vector machine (SVM). Recently, deep neural networks based methods have also been investigated. Chen *et al.* [11] proposed CNNs based framework for video deraining using superpixel segmentation. They aligned the scene contents at the superpixel-level, which improves robustness to rain occlusion and fast camera motion. By exploring the temporal redundancy in multiple images, Liu *et al.* [50] proposed a hybrid rain model to cover both rain streaks and occlusions. These methods make full use of the rich information in multiple images and the temporal

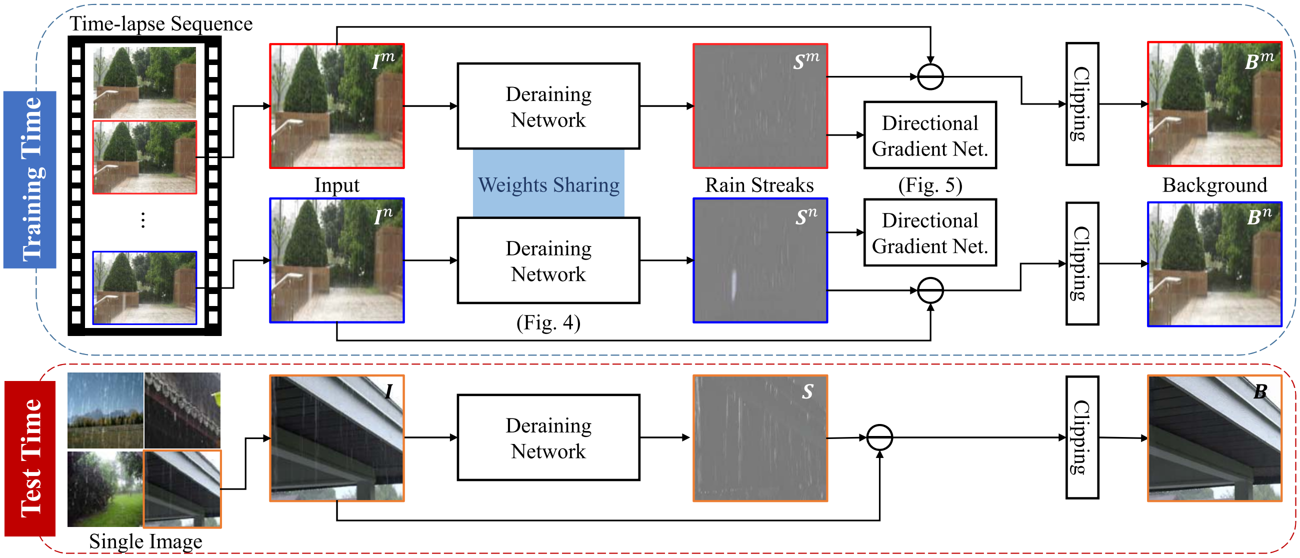


Fig. 2. The overview of the proposed learning framework. The proposed networks consist of two components: deraining networks and directional gradient networks. The deraining networks are trained to predict rain streaks and background images. The directional gradient networks are trained to determine a few dominant orientation of rain streaks. At inference time, only a single image is required.

redundancy in adjacent frames. While all the aforementioned methods require multiple images as inputs during both training and testing, our method only requires the time-lapse sequences which have spatially invariant background to train networks, and a single image at the testing phase.

C. Deraining Datasets

Existing rain datasets generated synthetically by commercial software such as Adobe After Effects [3], [5], [20], [22] have a limited realism. They cannot effectively reflect various real rain conditions such as rain shape, direction, and intensity. To alleviate this problem, Wang *et al.* [24] constructed a large-scale dataset of rain and clean image pairs that consists of natural rain scenes by leveraging temporal priors and human supervision. Our dataset is related to [24], but there are several key differences that put a significant gap between the two approaches. While [24] explicitly generated training data, *i.e.,* paired training data using percentile filtering and attention map through explicit supervision, our method builds the time-lapse sequences from natural rain scenes without clean images.

D. Using Multiple Images

To overcome the lack of training data in various computer vision and image processing tasks, numerous approaches [31], [32], [51]–[55] leveraged large amounts of multiple images or image sequences. For instance, Godard *et al.* [52] proposed a self-supervised learning approach for monocular depth estimation using stereo image pairs. Ma *et al.* [32] proposed a CNN-based intrinsic image decomposition using time-lapse datasets, where a deep network is trained with only multiple images containing same albedo but different shading. Vondrick *et al.* [54] proposed a video colorization for visual tracking by using large amounts of unlabeled video. Nam *et al.* [31] proposed a multi frame joint conditional generation framework for synthesizing a time-lapse video and photo-realistic illumination changes from a single outdoor image. A large-amounts of multiple images or image sequences typically contain a rich information between the coherent frames. Inspired by these approaches, we attempt to address the lack of real training data in the single image deraining task by leveraging the time-lapse sequences.

III. Proposed Method

A. Motivation and Overview

Let us denote an image degraded by rainy artifacts as I. It can be generally modeled as a summation of a rain streak S and a background B [6], [7], [15]–[19] such that

$$I = S + B. \tag{1}$$

The objective of single image deraining is to decompose I into the rain streak S and the background B [6], [18], [24], [39].

To this end, most CNN-based methods [3], [5], [20]–[22], [24], [40], [58] learn a mapping function between the rainy image I and the background B (or the rain streak S) with a large-scale training data consisting of rainy images and ground truth background (or rain streak images) in a supervised manner. Obtaining such data in real environments is, however, practically impossible, and thus they usually leverage the synthetic data generated by Photoshop [27] or photo realistic rendering technique [28]. They have shown excellent results over existing handcrafted approaches, but they suffer from the domain adaptation issue [5], [22] when applied to real rainy images.

To solve this limitation, we present a novel learning framework for single image deraining that leverages the time-lapse data. We present a background consistency loss that enables our deraining networks to consistently generate the same derained images from the time-lapse sequences, as shown in Fig. 2. We train the deraining networks using a set of time-lapse sequences $\mathbf{T} = \{T_c\}_{c=1,\dots,C}$, where $T_c = \{I^{c,1}, \dots, I^{c,N}\}$. N is the number of frames and c denotes the index of scenes, and C is the total number of scene.

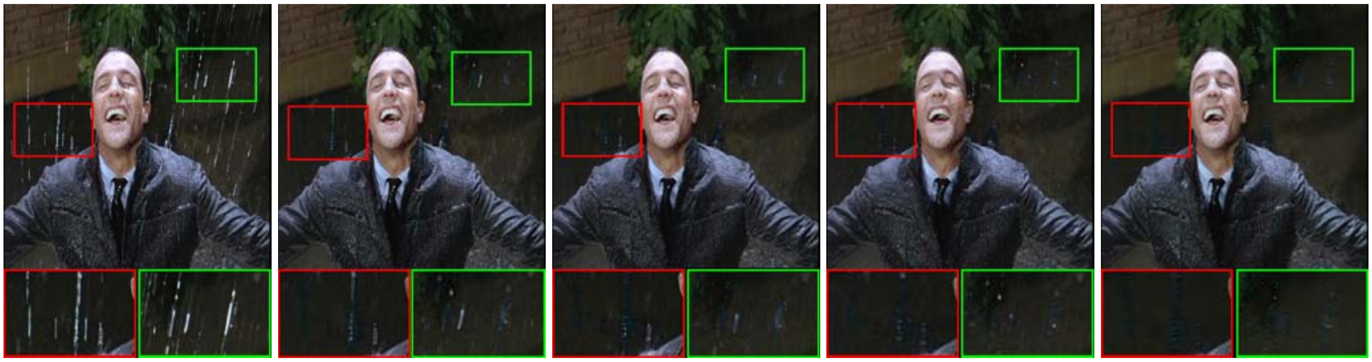


Fig. 3. Comparison of our loss functions: (a) input real rainy image, results of our method trained with (b) $\mathcal{L}_b$ only, (c) $\mathcal{L}_b$ and $\mathcal{L}_s$, (d) $\mathcal{L}_b$, $\mathcal{L}_s$, and $\mathcal{L}_r$, and (e) $\mathcal{L}_b$, $\mathcal{L}_r$, $\mathcal{L}_s$, and $\mathcal{L}_v$. By jointly using all the proposed loss functions, our method can provide highly improved performance.

We further present additional loss functions such as a structure similarity loss to make the input and output images have the similar structural information, and a directional gradient loss to make the estimated rain streaks have a few dominant gradients. The reconstruction loss is used following the definition of the rain model.

In the deraining networks, we first extract the multi-scale features at different scales, and then fuse them through a learned fusion weight, where the optimal fusion weight is dynamically determined conditioned on input features. In the directional gradient networks, dominant rain directions are trained. The clipping layer prevents background images to be negative. To train the networks with rainy images only, we construct a large-scale time-lapse dataset, where each scene contains the same backgrounds but different rain streaks. Note that during an inference, the networks only require a single image as input. Unlike the video deraining methods [2], [11], [47]–[50] making full use of the information among frames spatially and temporally, we exploit the time-lapse sequences including the spatial invariance for single image deraining. Moreover, while our method only uses single image at testing thanks to the time-lapse sequences and our network, video deraining methods require multiple images as inputs.

B. Loss Functions

1) Background Consistency Loss: In the time-lapse sequences T_c taken at a static scene, background images should be invariant to time-varying rain streak changes. To generate consistent background images across the time-lapse sequences, we formulate the background consistency loss that uses an L_1 penalty among the estimated background images such that

$$\mathcal{L}_b = \sum_{c} \sum_{\{m,n\}\in N} \sum_{i} \left\| B_i^{c,m} - B_i^{c,n} \right\|_1, \tag{2}$$

where $B^{c,m}$ is a background that is decomposed from $I^{c,m}$, and m and n represent indexes of different input images from the time-lapse sequences. Here, $B_i^{c,m}$ and $B_i^{c,n}$ are the pixel elements from the image $B^{c,m}$ and $B^{c,n}$, respectively. However, since minimizing this loss function is under-constrained, we present additional loss functions to further constrain the output.

2) Structure Similarity Loss: We argue that most of the color or texture in estimated backgrounds should be well approximated by input images. To realize this, inspired by [63], [64], we present the structural similarity loss that encourages estimated backgrounds to be close to the input images. This loss helps to initialize the structure of the overall background information. We minimize an L_1 penalty of estimated backgrounds and input images such that

$$\mathcal{L}_s = \sum_{c} \sum_{\{m,n\}\in N} \sum_{i} \gamma \left\| I_i^{c,m} - B_i^{c,n} \right\|_1, \tag{3}$$

where γ is reduced linearly from 0.1 to 0.0001 during the first 30% of the training and then fixed. This loss function enables the networks to produce good initial results at the early training stages. Fig. 3(c) shows the validation of this loss. Note that an input image and an output background image should be selected in different samples (i.e., $m \neq n$). When training the networks with the pair of the same input image (i.e., $m = n$), the networks are unable to reduce rain streaks effectively. Different input images prevent this undesirable effect during training.

3) Directional Gradient Loss: To enforce the estimated rain streaks to have a few dominant gradient directions, we present a novel loss function that clusters the gradients of rain streaks into majority cluster centroids. We first extract the gradient orientation of the estimated rain streaks such that $\theta = \tan^{-1}(\nabla_y S / \nabla_x S)$, where ∇_x and ∇_y indicate the gradient of x- and y-directions, respectively. To estimate the directional gradient centers $\{c_k\}$, inspired by [65]–[67], we minimize the following objective function:

$$V(k) = \sum_{i} \alpha_k(\theta_i) \left\| \theta_i - c_k \right\|_1, \tag{4}$$

where θ_i is the gradient orientation at pixel i and c_k is k-th cluster center. $\alpha_k(\theta_i)$ denotes the membership of the gradient orientation θ_i to k-th cluster, defined as follows:

$$\alpha_k(\theta_i) = \frac{e^{W_k^T \theta_i + b_k}}{\sum_{k'} e^{W_{k'}^T \theta_i + b_{k'}}}, \tag{5}$$

where W_k and b_k are sets of trainable parameters for k-th cluster. To learn the cluster centroids and make the gradient directions of all pixels be concentrated on them, we minimize $V(k)$ with an L_1 penalty such that

$$\mathcal{L}_v = \sum_k \|V(k)\|_1. \tag{6}$$

Fig. 3 shows the effectiveness of directional gradient loss in capturing rain streaks.

4) Reconstruction Loss: Following the definition of the rain model [6], [7], [15]–[19], the input image I should be reconstructed with B and S. We present a reconstruction loss as follows:

$$\mathcal{L}_r = \sum_c \sum_{m \in N} \sum_i \left\| I_i^m - (B_i^m + S_i^m) \right\|_1. \tag{7}$$

This strongly prevents any deviation from Eq. 1. Empirically, this loss reaches close to 0 after 10% of training.

5) Total Loss: With all the aforementioned loss functions, the total loss function is formulated such that

$$\mathcal{L}_t = \lambda_b \mathcal{L}_b + \lambda_s \mathcal{L}_s + \lambda_v \mathcal{L}_v + \lambda_r \mathcal{L}_r, \tag{8}$$

where λ_b, λ_s, λ_v, and λ_r are weighting factors.

C. Network Architecture

The proposed method consists of two sub-networks, including the deraining networks to estimate a rain streak and the directional gradient networks to enforce the estimated rain streak to have a few dominant gradient directions.

1) Deraining Networks: We formulate the deraining networks as encoder, dynamic fusion module, and decoder, as illustrated in Fig. 4. Based on the intuition that rain artifacts can be encoded at multiple scales [5], [19], [22]–[24], [58], we present multiple encoders E_3, E_5, and E_7 consisting of convolution layers having kernel sizes of 3×3, 5×5, and 7×7, respectively, as shown in Table I.

The features from multiple encoders are fused to predict the rain streaks (S). Multiple features from multiple encoders have its own receptive field containing various spatial contextual information. However, a simple concatenation disregards the characteristic of each feature [62] and may provide a limited performance. To address this limitation, we introduce a dynamic fusion module to find an optimal fusion weight. The dynamic fusion module dynamically combines the output feature of E_3, E_5, and E_7, where the optimal fusion weight $W_E^{G,*}$ can be learned with respect to each input with an additional convolutional network using filter-generator [33] such that

$$W_E^{G,*} = G(E_3(I), E_5(I), E_7(I); W_E^G), \tag{9}$$

with W_E^G denotes the parameters of filter generator. Since $W_E^{G,*}$ is conditioned on input features, we can find more optimal fusion weights. The concatenated features are then convolved with the generated filter, and resulting features are fed into the decoder. We will verify the effectiveness of the dynamic fusion module compared to other fusion methods in Section IV.C.2.

In the decoder, the spatial resolution of the encoder feature is progressively enlarged through the sequences of deconvolution and convolution layers, as shown in Table II. Each layer is composed of 3×3 deconvolution and convolution layers followed by ReLU, and is connected to the encoder using skip connections. The deconvolution layer consists of the transposed convolution with fixed bilinear upsampling kernel. The decoder yields the same resolution output as an input image. In addition, the decoder output, denoted by S, is subtracted from the input image using a subtraction layer. A clipping layer is finally applied to the residual to prevent a final output, denoted by B, from being a negative.

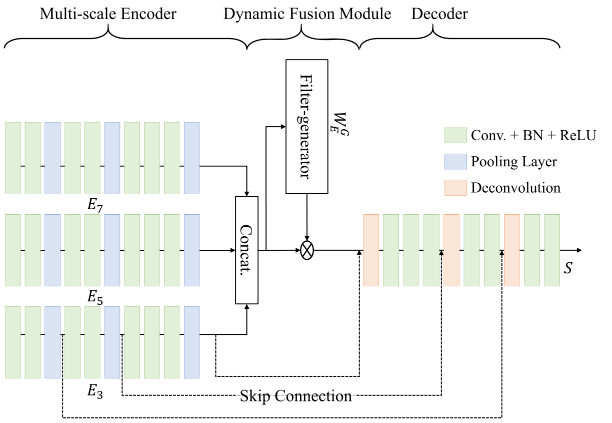


Fig. 4. The architecture of the deraining networks consisting of three components. The multi-scale encoder first captures features at different scales. The filter-generator aggregates multi-scale features adaptively. The decoder resolves spatial resolution details using skip connections.

TABLE I

NETWORK ARCHITECTURE OF THE MULTI-SCALE ENCODER OF DERAINING NETWORKS, WHERE 'KERNEL' REPRESENTS THE KERNEL SIZE OF CONVOLUTION LAYER, AND 'CH. I/O' AND 'DO. I/O' REPRESENTS CHANNELS AND DOWNSCALING FACTORS OF INPUT AND OUTPUT RELATIVE TO THE INPUT, RESPECTIVELY

	E_3 encoder		
Layer	Kernel	Ch. I/O	DO. I/O
conv1_1_3	3×3	1/64	1/1
conv1_2_3	3×3	64/64	1/1
max Pool1_3	2×2	64/64	1/2
conv2_1_3	3×3	64/128	2/2
conv2_2_3	3×3	128/128	2/2
max Pool2_3	2×2	128/128	2/4
conv3_1_3	3×3	128/256	4/4
conv3_2_3	3×3	256/256	4/4
conv3_3_3	3×3	256/256	4/4
Max Pool3_3	2×2	256/256	4/8
	E_5 encoder		
conv1_1_5	5×5	1/64	1/1
conv1_2_5	5×5	64/64	1/1
max Pool1_5	2×2	64/64	1/2
conv2_1_5	5×5	64/128	2/2
conv2_2_5	5×5	128/128	2/2
max Pool2_5	2×2	128/128	2/4
conv3_1_5	5×5	128/256	4/4
conv3_2_5	5×5	256/256	4/4
conv3_3_5	5×5	256/256	4/4
Max Pool3_5	2×2	256/256	4/8
	E_7 encoder		
conv1_1_7	7×7	1/64	1/1
conv1_2_7	7×7	64/64	1/1
max Pool1_7	2×2	64/64	1/2
conv2_1_7	7×7	64/128	2/2
conv2_2_7	7×7	128/128	2/2
max Pool2_7	2×2	128/128	2/4
conv3_1_7	7×7	128/256	4/4
conv3_2_7	7×7	256/256	4/4
conv3_3_7	7×7	256/256	4/4
Max Pool3_7	2×2	256/256	4/8

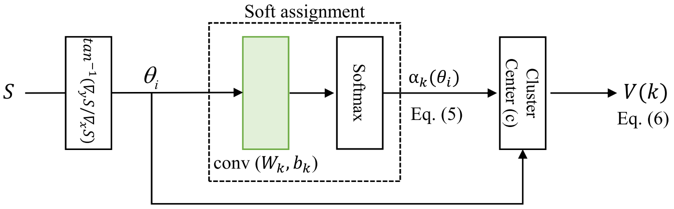


Fig. 5. Illustration of the directional gradient networks, implemented by standard CNN layers including convolutions and softmax.

TABLE II

NETWORK ARCHITECTURE OF THE FILTER-GENERATOR AND DECODER OF DERAINING NETWORKS, WHERE 'KERNEL' REPRESENTS THE KERNEL SIZE OF CONVOLUTION LAYER, AND 'CH. I/O' AND 'DO. I/O' REPRESENTS CHANNELS AND DOWNSCALING FACTORS OF INPUT AND OUTPUT RELATIVE TO THE INPUT, RESPECTIVELY. 'DYN' AND 'MK' MEANS DYNAMIC AND MAKE, RESPECTIVELY

Filter-generating Networks			
Layer	Kernel	Ch. I/O	DO. I/O
dyn_concat	-	256×3/768	8/8
mk_dyn_filter	1×1	768/768×256	8/8
dyn_filter	1×1	768/256	8/8
Decoder			
Layer	Kernel	Ch. I/O	DO. I/O
deconv3	3×3	256/256	8/4
concat3	-	256×2/512	4/4
conv3	3×3	1024/256	4/4
inv-conv3_3	3×3	256/256	4/4
inv-conv3_2	3×3	256/256	4/4
inv-conv3_1	3×3	256/128	4/4
deconv2	3×3	128/128	4/2
concat2	-	128×2/256	2/2
conv2	3×3	512/128	2/2
inv-conv2_2	3×3	128/128	2/2
inv-conv2_1	3×3	128/64	2/2
deconv1	3×3	64/64	2/1
concat1	-	64×2/128	1/1
conv1	3×3	256/64	1/1
inv-conv1_2	3×3	64/64	1/1
inv-conv1_1	3×3	64/1	1/1

2) Directional Gradient Networks: To regulate the estimated rain streak to have a few dominant gradient directions, we introduce the directional gradient networks, as shown in Fig. 5, where a few dominant gradient directions of rain streaks are trained as cluster centers [65]–[67]. The networks consist of convolution layers and a soft-max layer. The convolution layers consist of a set of k filters W_k that have spatial support 3 × 3 and biases b_k. The output of the convolution layers is passed through the soft-max function to obtain a soft assignment $\alpha_k(\theta_i)$ that weights the different terms in the cluster center layer. The weighted sum of θ_i and $\alpha_k(\theta_i)$ are trained in cluster center layer. By minimizing Eq. 6, we estimate W_k and b_k. Note that the directional gradient networks play the role of regularizing the deraining networks, and are not used during an inference.

D. Time-Lapse Sequence Dataset

1) Observation: Existing synthetic rain datasets have limited realism to model real rainy characteristics [24], [25]. There are some datasets [2], [11], [36], [50] generated synthetically by commercial software program such as Adobe After Effects[1], but they incur the domain adaptation problem.

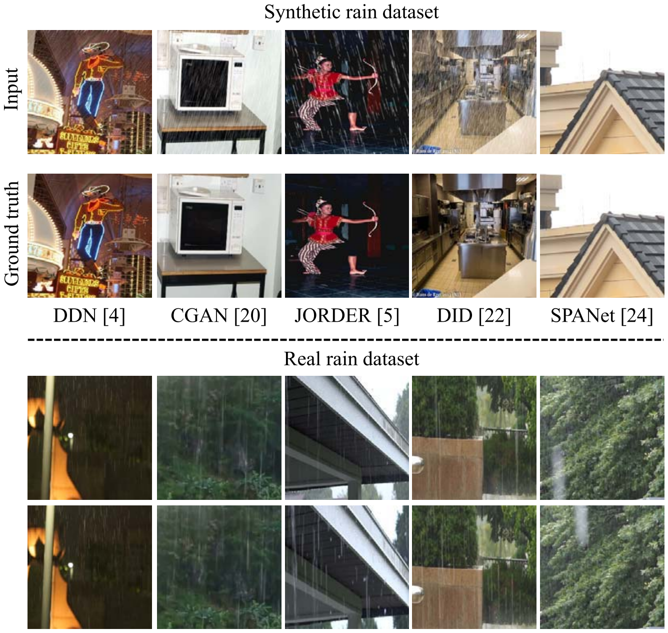


Fig. 6. Examples of synthetic rain dataset and our time-lapse sequences from real world. Our dataset contains more general rainy circumstances.

TABLE III

OVERVIEW OF THE EXISTING DATASETS AND OUR DATASET. *Total.*, *Train.*, AND *Test.* DENOTE THE TOTAL NUMBER OF DATASETS, TRAINING, TESTING IMAGES, RESPECTIVELY. '*' DENOTES THAT THE METHOD REQUIRES ADDITIONAL TRAINING DATA FOR TRAINING

Dataset	*Total.*	*Train.*	*Test.*	Additional data
Synthetic rain dataset				
DDN [4]	14,000	9,100	4,900	-
CGAN [20]	1,300	1,200	100	-
JORDER [5]*	2,200	2,000	200	Binary map
DID [22]*	13,200	12,000	1,200	Density level
Real rain dataset				
SPANet [24]*	29,500	28,500	1,000	Attention map
Ours	80,910	80,910	-	-

Instead, we construct the time-lapse sequences that enable our method to estimate background images through the structure preserving property [29]–[31]. Note that there were no attempts to use the time-lapse data to train the deraining networks directly.

2) Data Acquisition: We built up the training data consisting of the time-lapse sequences, each of which contains the same background scenes with different rain streaks, by taking our own real time-lapse data and collecting them from Youtube. For a better generalization, we considered various rain conditions at diverse scenes. We mounted a camera (Sony A7M2) on a tripod, and acquired 110 time-lapse sequences for outdoor scenes. 76 time-lapse sequences were also collected from Youtube. Since a rainy video, in which both camera and scenes are static over all frames, is rare among public videos, we extracted the static part of the rainy video. The time-lapse data was carefully examined to ensure that images in each set contain the various rain types in terms of shapes, directions, and sizes of rain streaks. Note that all the time-lapse sequences taken from real environments have no ground truth backgrounds. Fig. 6 and Table III show the comparison of existing

[1] https://www.adobe.com/AfterEffects

datasets and our dataset quantitatively and qualitatively. Our dataset improves the deraining performance effectively on real rainy images. The effectiveness of our dataset will be discussed in experiments.

E. Implementation Details

The proposed networks were implemented with the VLFeat MatConvNet library [68] library, using an NVIDIA GeForce GTX 1080 Ti GPU. All training images were cropped and then resized to 128 × 128 with a batch size of 4. We did not use data augmentation such as flipping and rotating because our data already contains sufficiently various scenes. For an efficient stochastic optimization, the Adam solver [69] was adopted with a fixed learning rate of 10^{-4} and momentum of 0.9. We set $k = 4$, $\lambda_b = 1$, $\lambda_s = 0.1$, $\lambda_v = 0.01$, and $\lambda_r = 0.001$. We set $C = 186$ and $N = 30$. For each rainy sequence, 2 images were sampled from 30 images, and thus total number of combinations is $186 \times {}_{30}C_2 = 80{,}910$. The E_3 encoder networks were the same architecture as the first 7 layers of VGG network [61]. Our method takes 2 days for training.

IV. Experiments

A. Experimental Settings

In experiments, we evaluated the proposed method in comparison to conventional hand-crafted approaches, such as discriminative sparse coding (DSC) [17], Gaussian mixture model (GMM) based method [18], joint convolutional analysis and synthesis sparse representation (JCAS) [34][2] and CNN based supervised approaches such as deep detailed network (DDN) [4],[3] joint rain detection and removal (JORDER) [5],[4] density-aware single image de-raining network (DID) [22],[5] non-locally enhanced encoder-decoder network [21] (NLEDN)[6] progressive image deraining networks [40] (PReNet),[7] Semi-supervised Transfer Learning for Image Rain Removal [25] (SIRR),[8] Spatial Attentive Single-Image Deraining [24] (SPANet),[9] JORDER-E [42], Depth-attentional Features for Single-image Rain Removal [14] (DAF-Net),[10] and Heavy Rain Image Restoration [56] (HeavyRain).[11] We used the pre-trained models provided by authors for comparison. Our method was trained with our time-lapse data using the dynamic fusion module and all loss functions.

For evaluation on real images, we use the SPANet [24] that provides 1,000 paired pseudo ground truth testset consisting of natural rain scenes using percentile filtering. We also use some examples collected from previous works [5], [20] and our dataset. We measure the performance of the synthesized data using two metrics, including Peak Signal-to-Noise Ratio (PSNR) and Structure Similarity Index (SSIM).

For synthetic data evaluation, we use three benchmark datasets, provided by DDN [4], JORDER [5], and DID [22]. DDN [4] provides 4900 rainy/clean image pairs, which were synthesized from 350 clean images with 14 different rain streaks. JORDER [5] provides Rain100H and Rain100L each of which consists of 100 images selected from BSD200 [70]. As pointed out in [5], [21], since the synthesized examples in Rain100H are inconsistent with real images, we used Rain100L for performance evaluation. The DID [22] provides 1,200 image pairs containing rain streaks with different orientations and scales, where 400 images are provided for each per rain density level (i.e., light, medium and heavy).

Furthermore, to evaluate the proposed method on the challenging scenarios that contain not only rainy but also haze degradations, we additionally collected the rainy images with haze from Internet.

B. Comparison With the State-of-the-Arts

1) Analysis on Real World Data: We first measured the deraining performance of all competing methods and ours on real rainy images. We collected real world dataset from previous works [5], [20], [24] and our dataset. Fig. 7 shows the qualitative evaluations on real rainy images. While existing methods [3], [5], [21], [22], [24], [25] suffer from artifacts on long and thin rain streaks, our method effectively removes various types of rain streaks and preserves background information well. The derained results on real rain images taken from different rain conditions and various scenes demonstrate the superiority of our method. Note that although the state-of-the-art methods [21], [24], [25], [40] achieve significant performance on synthetic datasets, their performance was limited to real rainy image. They had a difficulty in comprehensively considering the complex distribution of real rain since they could not generalize various type of rain perfectly. We also reported quantitative and qualitative comparisons on the SPANet data [24] as shown in Fig. 8 and Table IV. As shown in Table IV, the model-driven method such as JCAS [34] even outperforms some CNNs based methods, i.e., DDN [4] and DID [22]. Note that although the CNNs based methods are generally superior to handcrafted methods, they still suffer from generalization issue on real world data. For instance, PReNet [40] and SIRR [25] make some holes on rain regions (result in the first row in Fig. 8 (g) and result in the second row in Fig. 8 (h)). Unlike these, our method achieves an outstanding performance over existing state-of-the-art methods.

2) Analysis on Synthetic Data: We analyzed the supervised learning based methods [4], [5], [21], [22], [24], [25], [40] on various synthetic data. The comparison was summarized in Table V and Fig. 9. As shown in Table V, recent supervised learning based methods clearly outperform most hand-crafted methods [17], [18]. However, those trained deep models show limited performance on other synthetic datasets. Especially,

[2] https://sites.google.com/site/shuhanggu/home

[3] https://xueyangfu.github.io/projects/tip2017.html

[4] http://www.icst.pku.edu.cn/struct/Projects/joint_rain_removal.html

[5] https://github.com/hezhangsprinter/DID-MDN

[6] https://github.com/AlexHex7/NLEDN

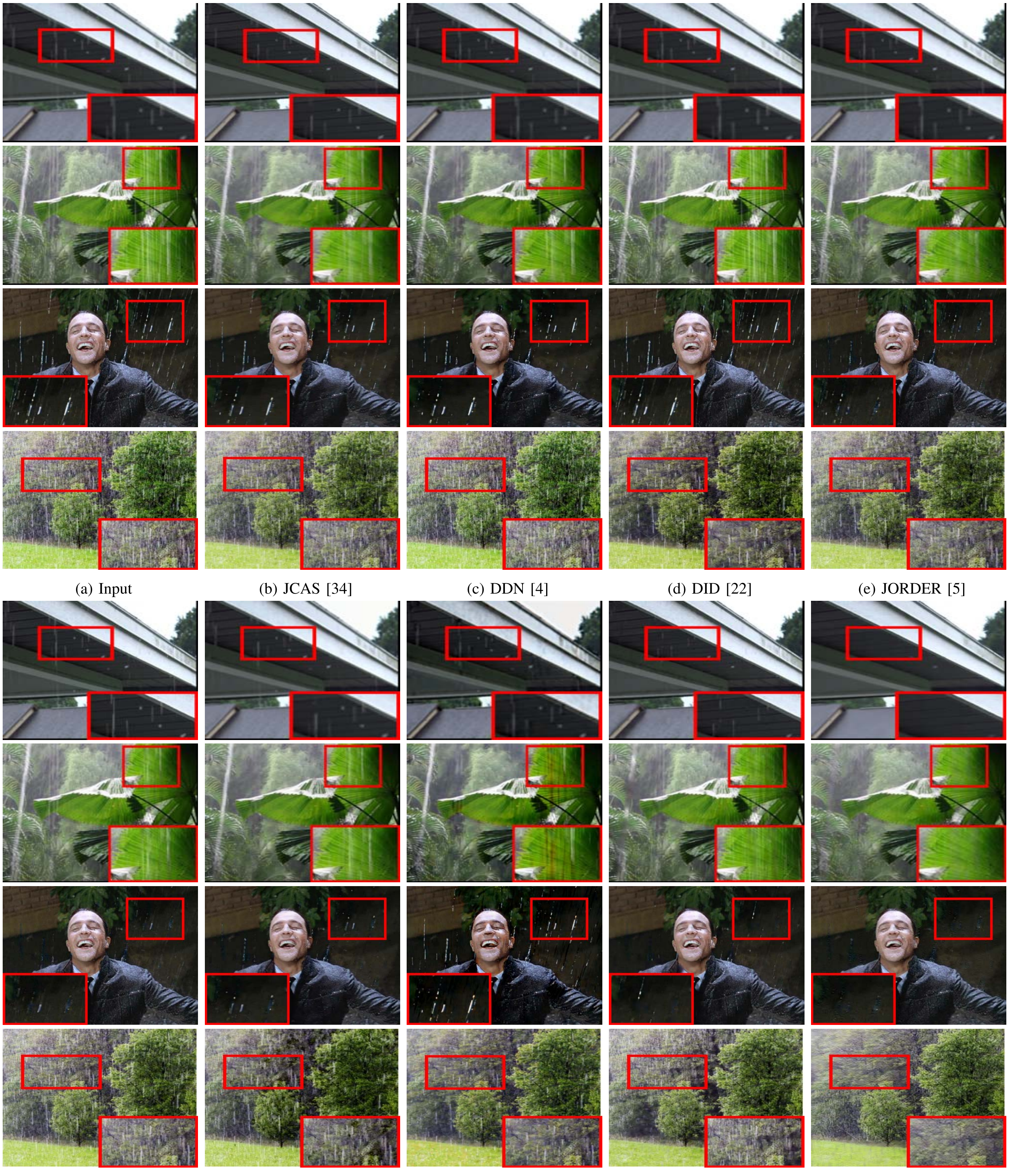
[7] https://github.com/csdwren/PReNet

[8] https://github.com/wwzjer/Semi-supervised-IRR

[9] https://stevewongv.github.io/derain-project.html

[10] https://github.com/xw-hu/DAF-Net

[11] https://github.com/liruoteng/HeavyRainRemoval



(a) Input (b) JCAS [34] (c) DDN [4] (d) DID [22] (e) JORDER [5]

(f) NLEDN [21] (g) PReNet [40] (h) SIRR [25] (i) SPANet [24] (j) Ours

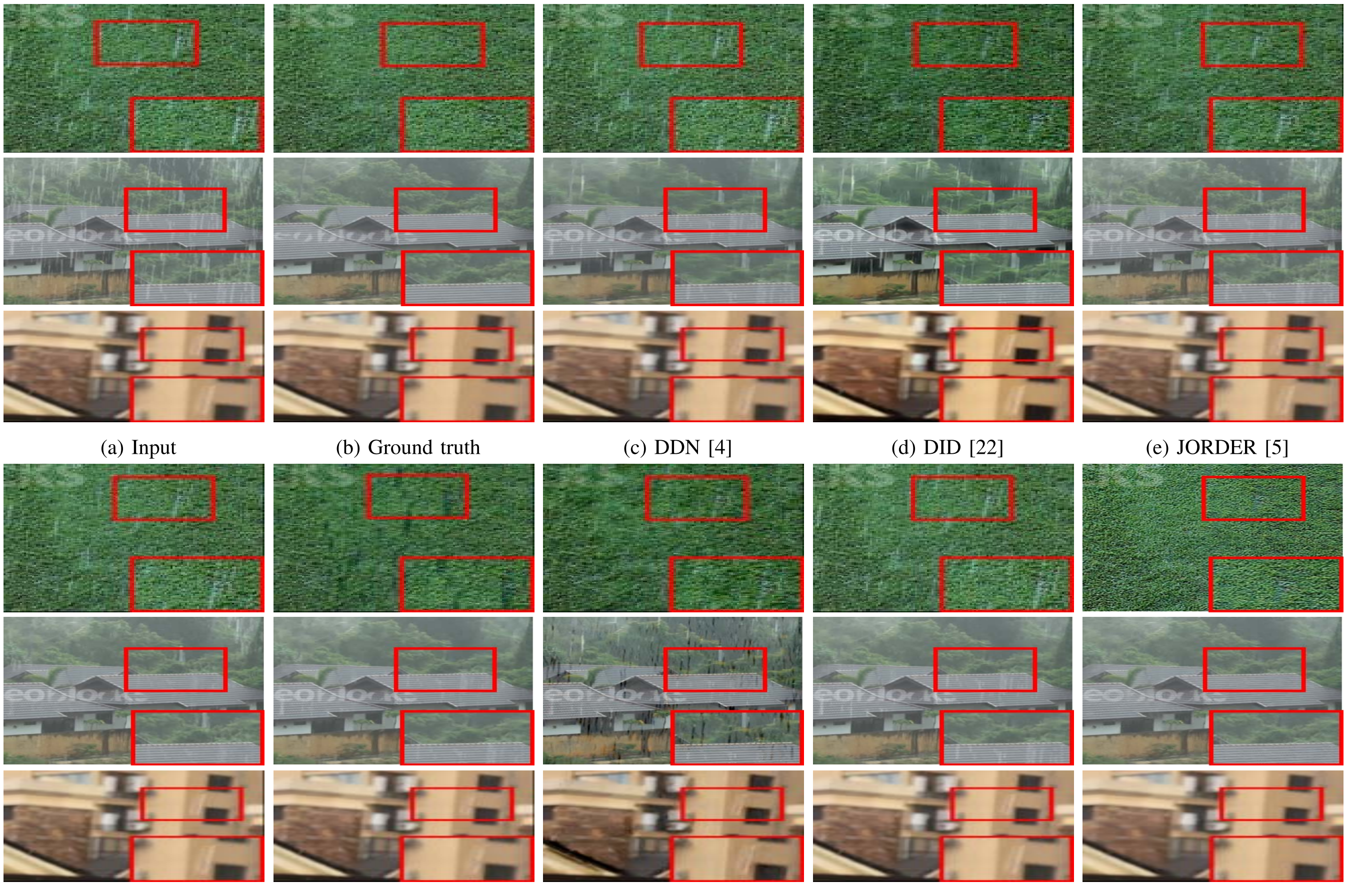
Fig. 7. Visual comparison of single image deraining on real rain images. Note that we collect real world examples from [5], [20] and our dataset.

TABLE IV

QUANTITATIVE COMPARISON OF THE STATE-OF-THE-ARTS AND PROPOSED METHOD ON SPANET DATA [24]

	DSC [17]	GMM [18]	JCAS [34]	DDN [4]	JORDER [5]	DID [22]	NLEDN [21]	PReNet [40]	SIRR [25]	SPANet [24]	Ours
PSNR	32.64	34.30	34.95	34.88	35.72	28.96	36.24	36.08	35.85	38.06	**38.54**
SSIM	0.932	0.943	0.945	0.973	0.977	0.941	0.980	0.978	0.972	0.987	**0.989**

due to a low generalization capability, JORDER [5] shows substantially degraded performance on DID [22] and DDN [4] test set. They still include rain streaks on JORDER [5] test set as exemplified in Fig. 9(c) and (d). Similar results have



(a) Input (b) Ground truth (c) DDN [4] (d) DID [22] (e) JORDER [5]

(f) NLEDN [21] (g) PReNet [40] (h) SIRR [25] (i) SPANet [24] (j) Ours

Fig. 8. Visual comparison of single image deraining on SPANet [24].

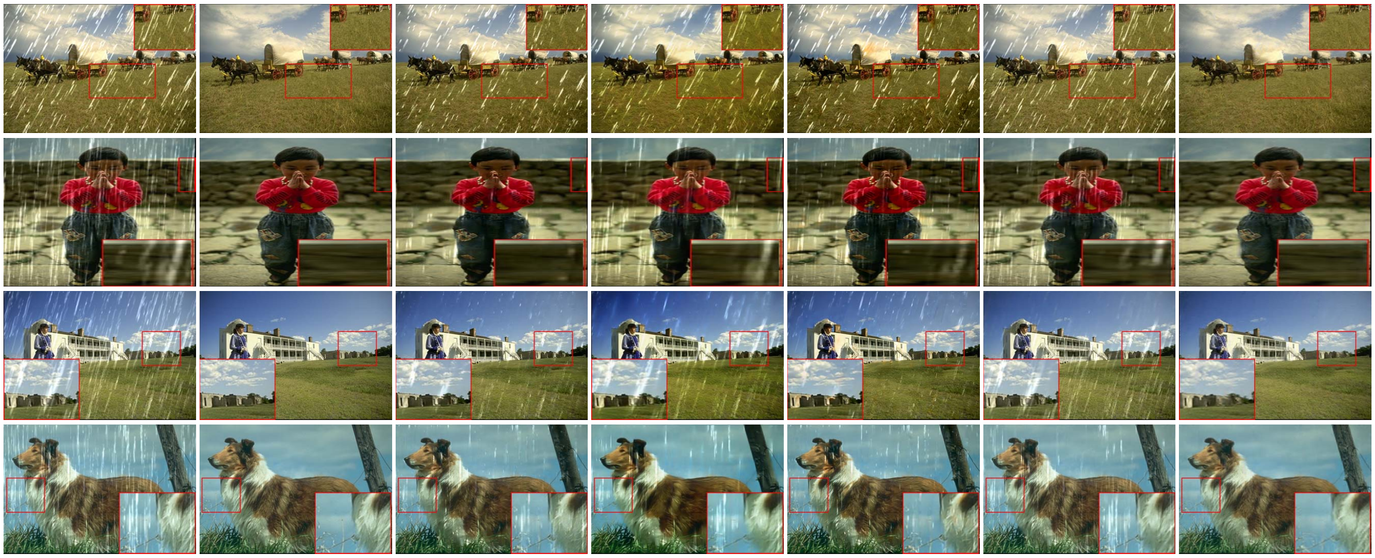
TABLE V

QUANTITATIVE COMPARISON OF SINGLE IMAGE DERAINING USING VARIOUS SYNTHETIC DATASET. GT MEANS THE METHOD USING PAIRED GROUND TRUTH DATA. '*', '**', AND '***' INDICATE THAT THE METHODS REQUIRE ADDITIONAL SUPERVISED CUE, I.E., BINARY MASK MAP, RAIN DENSITY LEVEL, AND ATTENTION MAPS RESPECTIVELY. THE HIGHER THE PSNR AND SSIM, THE BETTER

Benchmark	GT	Training data	Test set	PSNR	SSIM	Test set	PSNR	SSIM	Test set	PSNR	SSIM
DSC [17]	No	-	DDN	20.08	0.78	JORDER	23.39	0.86	DID	21.44	0.78
GMM [18]	No	-	DDN	20.66	0.81	JORDER	24.25	0.87	DID	22.75	0.83
DDN [4]	Yes	DDN	DDN	25.63	0.88	JORDER	25.99	0.81	DID	27.33	0.89
JORDER [5]*	Yes	JORDER	DDN	22.26	0.84	JORDER	35.23	0.96	DID	24.32	0.86
DID [22]**	Yes	DID	DDN	26.07	0.90	JORDER	30.48	0.93	DID	27.95	0.90
NLEDN [21]	Yes	JORDER	DDN	29.79	0.90	JORDER	36.57	0.95	DID	30.48	0.91
PReNet [40]	Yes	JORDER	DDN	32.55	**0.94**	JORDER	37.35	0.97	DID	31.20	0.90
SIRR [25]	Yes	DDN	DDN	28.44	0.89	JORDER	32.37	0.92	DID	28.44	0.89
SPANet [24]***	Yes	SPANet	DDN	29.76	0.90	JORDER	34.46	0.96	DID	28.76	0.90
Ours	No	Time-lapse	DDN	**33.73**	**0.94**	JORDER	**37.89**	**0.98**	DID	**33.25**	**0.93**

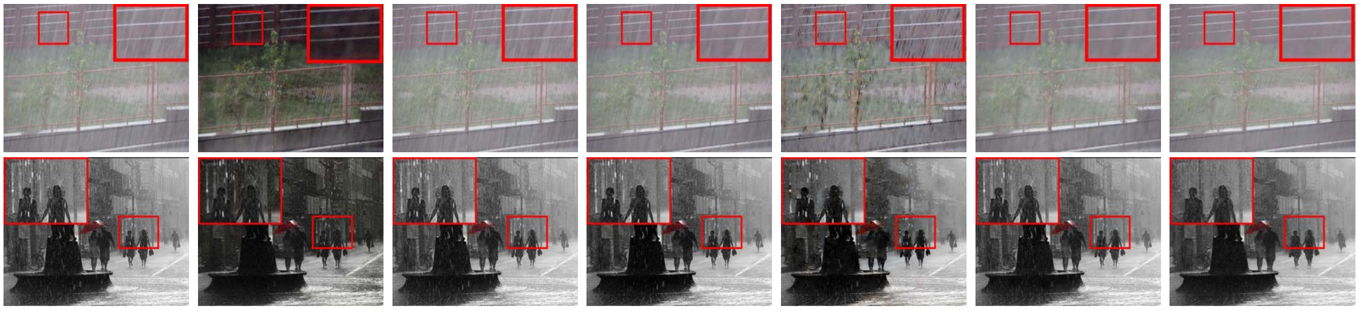
been shown from other methods and datasets, implying that the supervised learning based methods using specific synthetic data have limited generalization. The proposed method consistently achieves the best quantitative performance compared to other supervised methods as shown in Table V. The qualitative results in Fig. 9 show that our method generates plausible derained images at the synthetic data. It is noteworthy that although our network uses only time-lapse sequences without using any ground truth data, it outperforms the state-of-the-art supervised methods on synthetic dataset.

3) Analysis on Rainy With Haze: For more analysis, we conducted experiments on real rain images degraded by haze effects. For this, we additionally collected rain image degraded with haze from Internet. Then, we applied our network to estimate derained images, and compared with state-of-the-art methods including DAF-Net [14] which proposed a rain imaging model with rain streaks and haze. Fig. 10 shows our comparison results. The first row shows the derained results under long and thin rain accompanied with haze. While existing methods are difficult to handle the long diagonal



(a) Input (b) Ground truth (c) DDN [4] (d) DID [22] (e) SIRR [25] (f) SPANet [24] (g) Ours

Fig. 9. Results of single image deraining using synthetic JORDER test data [5]: (a) the input image, (b) ground truth image, (c) DDN [4], (d) DID [22], (e) SIRR [25], (f) SPANet [24], and (g) Ours.



(a) Input (b) DAF-Net [14] (c) DID [22] (d) NLEDN [21] (e) SIRR [25] (f) SPANET [24] (g) Ours

Fig. 10. Comparison results from the proposed method against those from the state-of-the-art methods on real rain images accompanied with haze.

rain streaks, our method is successful in removing the rain streaks even with haze. SIRR [25] generates the derained result corrupted the background. DAFNet shows haze removal effectively, however, it leaves some rain streaks. The second row shows that the existing methods fail to remove large and small rain streaks in the light haze. Though DAF-Net could handle haze removal, it is still challenging in rain removal. Unlike those, our method could remove rain streaks affected by haze in real rainy image thanks to our time-lapse sequences acquired in the real world and composed of various real circumstances.

TABLE VI

QUANTITATIVE COMPARISON OF THE PROPOSED METHOD TRAINED WITH VARIOUS LOSS FUNCTIONS WITH AND WITHOUT DYNAMIC FUSION MODULE. DYN. DENOTES THE DYNAMIC FUSION MODULE

$\mathcal{L}_r$	$\mathcal{L}_b$	$\mathcal{L}_s$	$\mathcal{L}_v$	Dyn.	PSNR	SSIM
✓	✓				37.68	0.981
✓	✓	✓			37.92	0.985
	✓	✓	✓		38.01	0.986
✓	✓	✓	✓		38.11	0.986
✓	✓	✓		✓	38.29	0.987
	✓	✓	✓	✓	38.46	0.988
✓	✓	✓	✓	✓	38.54	0.989

C. Ablation Study

We conducted an ablation analysis on different components and loss functions in our framework. For the quantitative evaluation, we used the test split of the SPANet [24].

1) Analysis of Loss Functions: Using time-lapse sequences from our dataset, we evaluated the effectiveness of the proposed loss functions, including reconstruction loss, structure similarity loss, background consistency loss, and directional gradient loss. In Table VI, we start from the our model trained with $\mathcal{L}_r$ and $\mathcal{L}_b$, and sequentially add other components. At first, we compare the model trained with and without structure similarity loss $\mathcal{L}_s$. We show that the structure similarity loss improves the deraining results. This loss function aids the deraining network to estimate some background information coming from I. The second and third row show that the model trained with directional gradient loss achieves much better intermediate results than the model trained with reconstruction loss. The deraining result by directional gradient loss is also visually more plausible, as shown in Fig. 11. In Fig. 11(b) and (c), the deraining result trained with directional gradient loss finds main direction of rain streaks, and removes rain streaks more precisely, demonstrating that finding main

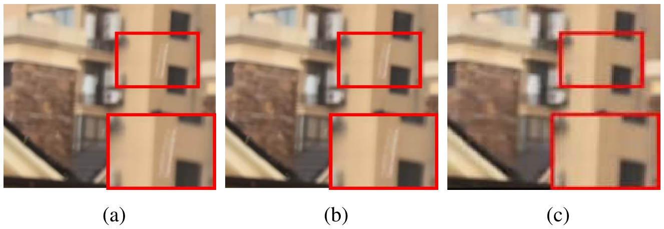


Fig. 11. Comparison of the deraining networks trained without and with directional gradient loss function: (a) input image, results of our method trained (b) without and (c) with directional gradient loss function. Our deraining networks with directional gradient loss function more effectively remove rain streaks.

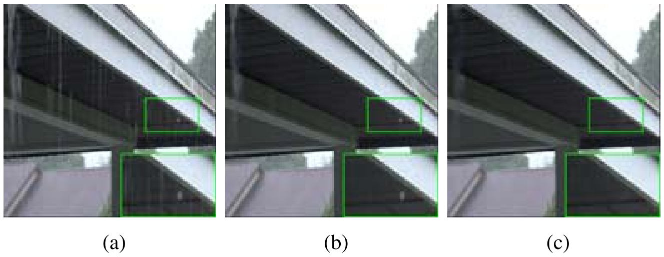


Fig. 12. Performance gain by dynamic fusion module: (a) input image, results of our method trained (b) without and (c) with dynamic fusion module. Our deraining networks with dynamic fusion module more effectively remove rain streaks.

TABLE VII

QUANTITATIVE COMPARISON OF THE PROPOSED METHOD TRAINED WITH DIFFERENT FUSION METHODS. CONCAT., SUM., PROD. AND DYN. DENOTES THE CONCATENATION, SUMMATION, PRODUCTION, AND DYNAMIC FUSION MODULE, RESPECTIVELY

	Sum.	Prod.	Concat.	Dyn.
PSNR	37.98	38.02	38.11	38.54
SSIM	0.983	0.984	0.986	0.989

directional gradient orientation helps deraining. We show that the results of the model trained with all loss functions achieves the highest PSNR and SSIM.

2) Analysis of Dynamic Fusion Module: We also evaluated the performance of the dynamic fusion module. Table VI shows that the model trained with the dynamic fusion module achieves a substantial accuracy gain over the model trained without dynamic fusion module. The effectiveness of the proposed dynamic fusion module is also shown in Fig. 12.

Moreover, we compare the deraining performances of our networks with several fusion methods including summation, product, concatenation, and dynamic fusion. The summation and product fusion produce the fused features by element-wise summation and multiplication, respectively. The concatenation fusion concatenates the features in the channel dimension. All models are trained with the total loss (Eq. (8)). The results are quantitatively given in Table VII. Since the dynamic fusion module dynamically learns optimal fusion weight conditionally determined by multiple features, the model with the dynamic fusion results achieves the best performance.

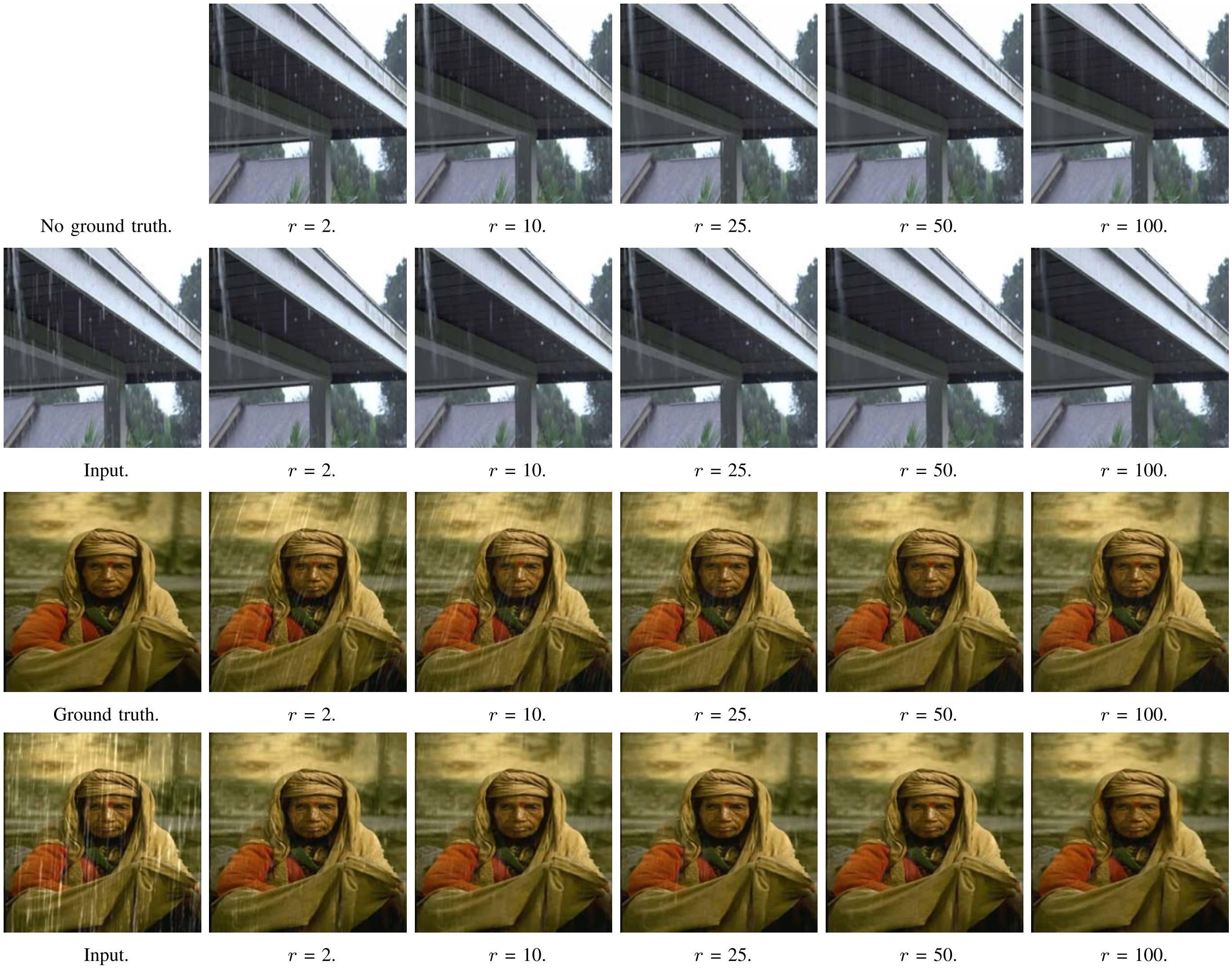
3) Compared With Time-Averaging: To verify the effectiveness of our framework using the time-lapse data as weak supervisions, we compared it with a simple time-averaging operation. Considering the characteristics of the time-lapse sequences taken from a static scene, simply averaging all frames along a time may reduce an interference by rain streaks. The time-averaging result can be used as a final derained image, but this approach is not practical as it always requires using multiple frames for performing the deraining task. Contrarily, our networks are trained using the time-lapse sequence, but only a single input image is used during an inference. Alternatively, the time-averaging results can play a role of supervisions in training the proposed networks. Namely, the proposed networks can be trained in a supervised manner with a pair of input image and the time-averaged output. In this case, the upper bound of the supervised learning approach is determined by the accuracy of supervision used for training. Fig. 13 and 14 show the qualitative and quantitative results of the proposed method and the time-averaging operation.

In Fig. 13, our results show that the more images are used, the better the performance. In contrast, the time-averaged results still contain rain streaks patterns. Fig. 14 shows that when the number of input images with different rain streaks is small, the proposed method achieves better performance than the time-averaging operation. Even when the number of rain streaks in training data increases, the proposed method still outperforms the time-averaging operation in terms of PSNR. This indicates that our approach using the time-lapse sequences as weak supervisions is a much better choice than the supervised approach using time-averaged outputs as the pseudo ground truth.

4) Analysis of Learned Features: To better understanding what the networks encode, we provide the visualization of learned features by our network. Fig. 15 shows a real rain image, our results, and feature maps of the first and last convolution layers. Fig. 15(d) shows four intermediate features of the convolutional output of input rain image in encoder of the first convolution layers. These contain the various types of rain streaks, and object edges which are uncorrelated to the rain streaks (i.e., the details of trees and grass). Fig. 15(e) shows four intermediate features of the last decoder layer. This feature maps show highly correlated with rain streaks. The visualization of learned features demonstrates that the deraining network discriminates and removes rain streaks and background effectively.

D. Analysis of the Number of Network Parameters

We conducted experiments to analyze the effects of the number of parameters in our method. By reducing the number of parameters, i.e., the kernel size of convolution layer from 3 × 3, 5 × 5, and 7 × 7 to 1 × 1, 3 × 3, and 5 × 5, and the number of input and output channels for each layer, we measured our performance for single image deraining. The results are shown in Fig. 17. Surprisingly, even with the small enough parameters, our method still has shown competitive performances compared to other methods. Furthermore, the performance gap as varying the network



No ground truth. r = 2. r = 10. r = 25. r = 50. r = 100.

Input. r = 2. r = 10. r = 25. r = 50. r = 100.

Ground truth. r = 2. r = 10. r = 25. r = 50. r = 100.

Input. r = 2. r = 10. r = 25. r = 50. r = 100.

Fig. 13. Visual comparison of time-averaged results (first and third rows) and our results (second and fourth rows) on real (first and second rows) and synthetic (third to fourth rows) time-lapse data. Note that r is the number of input images that contains different rain streaks.

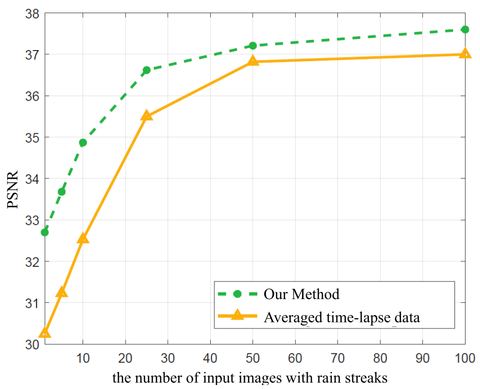


Fig. 14. Quantitative evaluation of the proposed method and averaged time-lapse sequences on the JORDER [5] dataset (Rain 100L).

parameters was so marginal. We guess those networks were already having high capacity to contain the rainy artifacts, and the high performances are attributed to the proposed loss functions.

E. Application on High-Level Tasks

Existing single image deraining methods focused mainly on training their models on certain type of synthetic images and then validating their methods on synthetic data and a few real images [13]. In this section, we explore how effective the proposed deraining method is as a preprocessing step for high-level tasks. We applied off-the-shelf semantic segmentation method [71] on the derained results. Since there are no rainy images with ground truth segmentation maps, we visualized only qualitative results. As shown in Fig. 16, our derained image is beneficial compared to the input rainy image and derained results obtained from state-of-the-arts methods [4], [5], [22], [24], [25], e.g., in road region and traffic sign.

We further conducted the experiments for studying the problem of object detection in rain images. Fig. 18 shows a visual result of object detection by applying the off-the-shelf object detection algorithm [72]. Since rain steaks cause blur and occlude background scenes, we expect that the

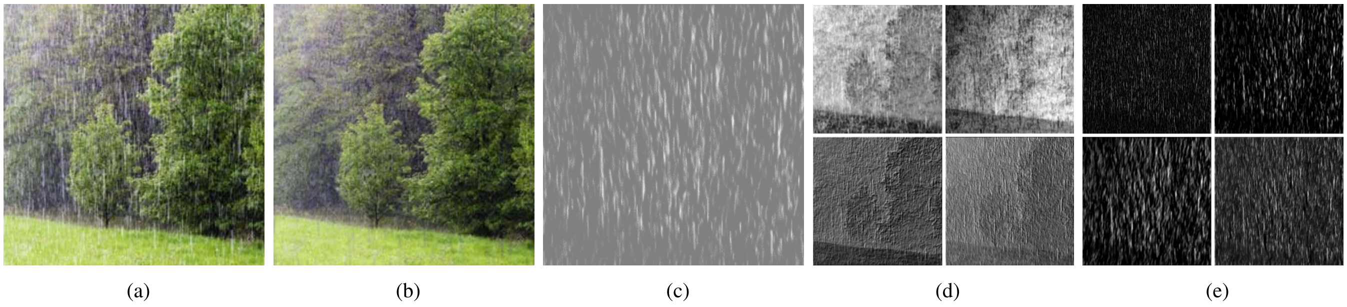


Fig. 15. Visualization of deraining and intermediate results on real rain image. (a) input rain image, (b) deraining results, (c) estimated rain streaks corresponding (b), (d) feature maps from the encoder of the first convolution layers, and (e) feature maps from the decoder of the last convolution layers.

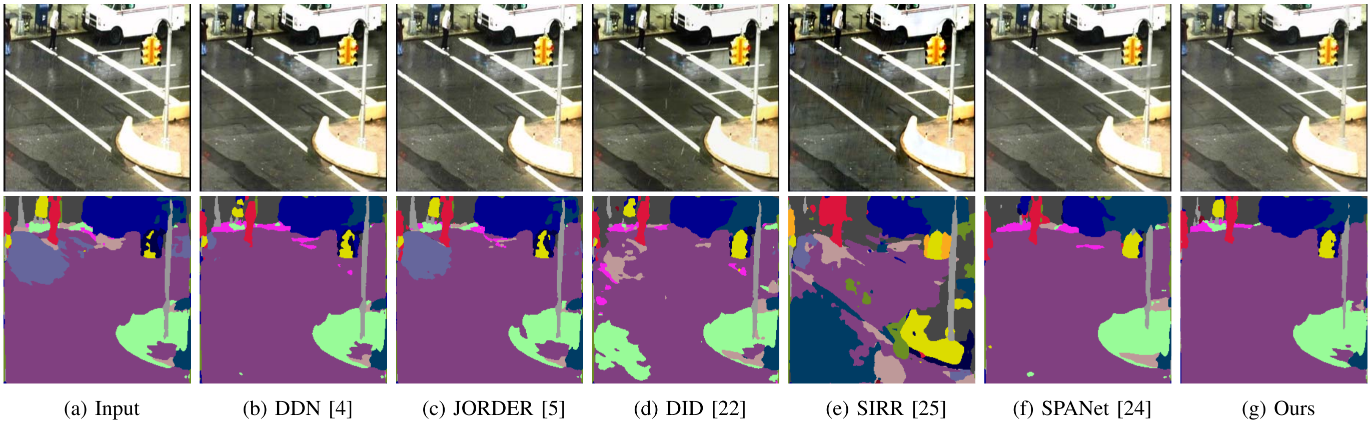


Fig. 16. Visualization of deraining results and semantic segmentation on the deraining results: (a) input rain image, deraining and semantic segmentation results using (b) DDN [4], (c) JORDER [5], (d) DID [22], (e) SIRR [25], (f) SPANet [24], and (g) Ours.

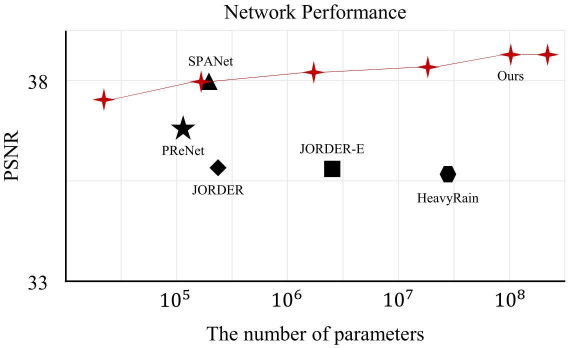


Fig. 17. Quantitative comparison of single image deraining on SPANet [24] according to the number of parameters.

performance of object detection will degrade in rainy circumstances. It is obviously that rain streaks can degrade the performance of object detection, i.e., by missing detections and producing low recognition confidence. In contrast, our derained results show that the detection performance has a significantly improvement over the baseline object detection algorithm.

F. Running Time

Table VIII shows the running time comparisons of our method and existing methods. We follow the original setting of all the released codes. On average, our method takes about 0.39s to obtain derained image of size 512 × 512.

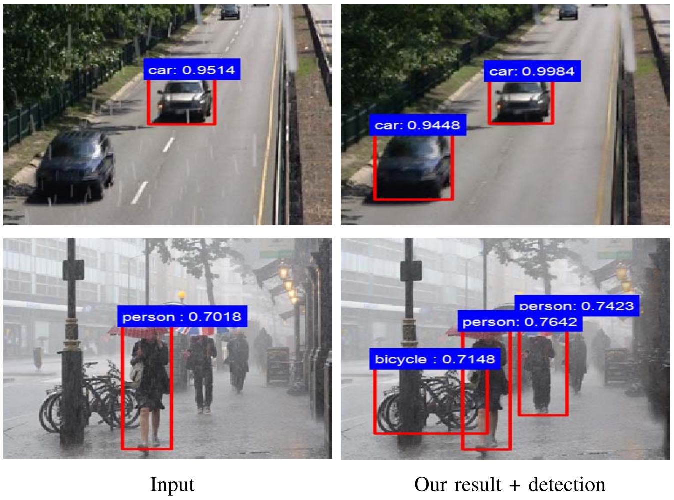


Fig. 18. Visualization comparison of object detection with and without deraining.

TABLE VIII

AVERAGED PSNR AND SSIM VALUE ON SYNTHESIZED IMAGES WITH THEIR COMPUTATIONAL TIME (SECOND). WE AVERAGED ON 1000 IMAGES WITH SIZE 512 × 512

	JORDER [5]	PReNet [40]	SIRR [25]	SPANet [24]	Ours
Time	0.18	0.08	0.71	0.11	0.39

G. Failure Cases

Even though our method achieves an outstanding performance on various rain conditions, we found that our model

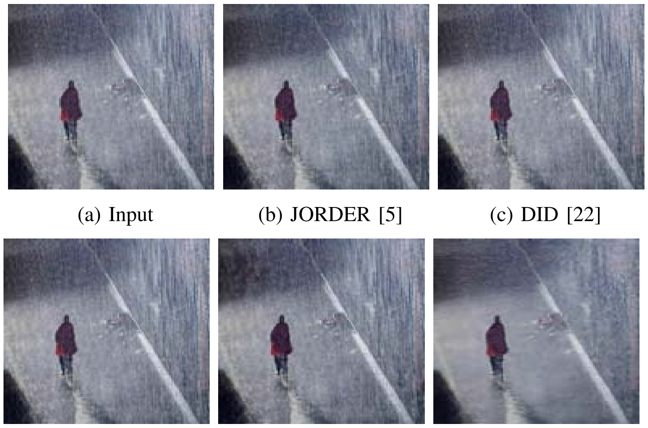


Fig. 19. Failure cases: similar to other methods, our method fails to generate derained images on extremely heavy rain conditions.

often failed to generate the derained image under heavy rain conditions, as shown in Fig. 19. It is difficult to get clean information from the heavy rainy image, and thus our results are unsatisfactory and blurry results. However, our method still outperformed the state-of-the-art methods.

V. Conclusion

We have introduced a novel learning framework to train single image deraining networks using the time-lapse dataset. Using the observation that multiple rainy images taken at a static scene have consistent backgrounds, we presented the background consistency loss to enforce the estimated background images to be similar. A novel structural similarity loss has been proposed to ensure that input and output images have similar structural information. For the estimated rain streaks image, we further introduced the directional gradient loss to make the estimated rain streaks have the main directional gradients. The dynamic fusion module was presented to effectively fuse multi-scale features in the deraining networks. Experiments have shown that our method is superior to state-of-the-arts methods and generalizes well on real rainy environments. In future work, we will investigate the deraining in an unsupervised way.

References

[1] X. Zheng, Y. Liao, W. Guo, X. Fu, and X. Ding, “Single-image-based rain and snow removal using multi-guided filter,” in *Proc. Int. Conf. Neural Inf. Process.*, Nov. 2013, pp. 258–265.

[2] J. Kim, J. Sim, and C. Kim, “Video deraining and desnowing using temporal correlation and low-rank matrix completion,” *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2658–2670, May 2015.

[3] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, “Clearing the skies: A deep network architecture for single-image rain removal,” *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2944–2956, Jun. 2017.

[4] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, “Removing rain from single images via a deep detail network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1715–1723.

[5] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, “Deep joint rain detection and removal from a single image,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1357–1366.

[6] L. Zhu, C.-W. Fu, D. Lischinski, and P.-A. Heng, “Joint bi-layer optimization for single-image rain streak removal,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2526–2534.

[7] Y. Chang, L. Yan, and S. Zhong, “Transformed low-rank model for line pattern noise removal,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1726–1734.

[8] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[9] T. Song, Y. Kim, C. Oh, and K. Sohn, “Deep network for simultaneous stereo matching and dehazing,” in *Proc. Brit. Mach. Vis. Conf.*, 2018, p. 5.

[10] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang, “DesnowNet: Context-aware deep network for snow removal,” *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 3064–3073, Jun. 2018.

[11] J. Chen, C.-H. Tan, J. Hou, L.-P. Chau, and H. Li, “Robust video content alignment and compensation for rain removal in a CNN framework,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6286–6295.

[12] Z. Fan, H. Wu, X. Fu, Y. Huang, and X. Ding, “Residual-guide network for single image deraining,” in *Proc. ACM Multimedia Conf. Multimedia Conf. (MM)*, 2018, pp. 1751–1759.

[13] S. Li *et al.*, “Single image deraining: A comprehensive benchmark analysis,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3838–3847.

[14] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng, “Depth-attentional features for single-image rain removal,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8022–8031.

[15] D.-A. Huang, L.-W. Kang, M.-C. Yang, C.-W. Lin, and Y.-C.-F. Wang, “Context-aware single image rain removal,” in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2012, pp. 164–169.

[16] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, “Automatic single-image-based rain streaks removal via image decomposition,” *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012.

[17] Y. Luo, Y. Xu, and H. Ji, “Removing rain from a single image via discriminative sparse coding,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3397–3405.

[18] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, “Rain streak removal using layer priors,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2736–2744.

[19] R. Li, L.-F. Cheong, and R. T. Tan, “Single image deraining using scale-aware multi-stage recurrent network,” 2017, *arXiv:1712.06830*. [Online]. Available: http://arxiv.org/abs/1712.06830

[20] H. Zhang, V. Sindagi, and V. M. Patel, “Image de-raining using a conditional generative adversarial network,” 2017, *arXiv:1701.05957*. [Online]. Available: http://arxiv.org/abs/1701.05957

[21] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin, “Non-locally enhanced encoder-decoder network for single image de-raining,” in *Proc. ACM Multimedia Conf. Multimedia Conf. (MM)*, 2018, pp. 1056–1064.

[22] H. Zhang and V. M. Patel, “Density-aware single image de-raining using a multi-stream dense network,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 695–704.

[23] M. Li *et al.*, “Video rain streak removal by multiscale convolutional sparse coding,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6644–6653.

[24] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. H. Lau, “Spatial attentive single-image deraining with a high quality real rain dataset,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12270–12279.

[25] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, “Semi-supervised transfer learning for image rain removal,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3877–3886.

[26] X. Jin, Z. Chen, J. Lin, Z. Chen, and W. Zhou, “Unsupervised single image deraining with self-supervised constraints,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2761–2765.

[27] [Online]. Available: https://www.photoshopessentials.com/photo-effects/rain/

[28] K. Garg and S. K. Nayar, “Photorealistic rendering of rain streaks,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 996–1002, Jul. 2006.

[29] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[30] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[31] S. Nam, C. Ma, M. Chai, W. Brendel, N. Xu, and S. J. Joo Kim, “End-to-end time-lapse video synthesis from a single outdoor image,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1409–1418.

[32] W. Ma, H. Chu, B. Zhou, R. Urtasun, and A. Torralba, “Single image intrinsic decomposition without a single intrinsic image,” in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 201–217.

[33] X. Jia, B. De, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 667–675.

[34] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1708–1716.

[35] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4057–4066.

[36] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "FastDeRain: A novel video rain streak removal method using directional gradient priors," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 2089–2102, Apr. 2019.

[37] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1968–1975.

[38] H. Zhang and V. M. Patel, "Convolutional sparse and low-rank coding-based rain streak removal," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2017, pp. 1259–1267.

[39] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 254–269.

[40] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3937–3946.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[42] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 6, pp. 1377–1393, Jun. 2020.

[43] W. Yang, J. Liu, S. Yang, and Z. Guo, "Scale-free single image deraining via visibility-enhanced recurrent wavelet learning," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2948–2961, Jun. 2019.

[44] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight pyramid networks for image deraining," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 1794–1807, Jun. 2020.

[45] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2004, p. 1.

[46] K. Garg and S. K. Nayar, "When does a camera see rain?" in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2005, pp. 1067–1074.

[47] K. Garg and S. K. Nayar, "Vision and rain," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 3–27, Jul. 2007.

[48] X. Zhang, H. Li, Y. Qi, W. Leow, and T. Ng, "Rain removal in video by combining temporal and chromatic properties," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 461–464.

[49] P. Barnum, T. Kanade, and S. Narasimhan, "Spatio-temporal frequency analysis for removing rain and snow from videos," in *Proc. Workshop Photometric Anal. Comput. Vis.*, 2007.

[50] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? Deep joint recurrent rain removal and reconstruction in videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3233–3242.

[51] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 577–593.

[52] C. Godard, O. M. Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 270–279.

[53] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1851–1858.

[54] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy, "Tracking emerges by colorizing videos," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 391–408.

[55] D. J. Rezende, S. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess, "Unsupervised learning of 3d structure from images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4996–5004.

[56] R. Li, L.-F. Cheong, and R. T. Tan, "Heavy rain image restoration: Integrating physics model and conditional adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1633–1642.

[57] PH. Seo, Z. Lin, S. Cohen, X. Shen, and B. Han, "Progressive attention networks for visual attribute prediction," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–19.

[58] R. Yasarla and V. M. Patel, "Uncertainty guided multi-scale residual learning-using a cycle spinning CNN for single image de-raining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8405–8414.

[59] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.

[60] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[61] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[62] S. Kim, S. Kim, D. Min, and K. Sohn, "LAF-net: Locally adaptive fusion networks for stereo confidence estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 205–214.

[63] L. Lettry, K. Vanhoey, and L. Van Gool, "Unsupervised deep single-image intrinsic decomposition using illumination-varying image sequences," *Comput. Graph. Forum*, vol. 37, no. 7, pp. 409–419, Oct. 2018.

[64] L. Lettry, K. Vanhoey, and L. Van Gool, "Deep unsupervised intrinsic image decomposition by siamese training," 2018, *arXiv:1803.00805v1*. [Online]. Available: https://arxiv.org/abs/1803.00805v1

[65] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5297–5307.

[66] S. Kmiec, J. Bae, and R. An, "Learnable pooling methods for video classification," 2018, *arXiv:1810.00530*. [Online]. Available: http://arxiv.org/abs/1810.00530

[67] S. Bhardwaj, M. Srinivasan, and M. M. Khapra, "Efficient video classification using fewer frames," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019.

[68] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, 2015, pp. 689–692.

[69] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[70] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Jul. 2001, pp. 416–423.

[71] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1925–1934.

[72] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

Jaehoon Cho (Student Member, IEEE) received the B.S. degree in electronic engineering and avionics from Korea Aerospace University, Gyeonggi, South Korea, in 2016. He is currently pursuing joint M.S. and Ph.D. degrees in electrical and electronic engineering with Yonsei University. His current research interests include deep-learning-based image processing, particularly in the area of bad weather restoration, related applications, and theories.

Seungryong Kim (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea, in 2012 and 2018, respectively. From 2018 to 2019, he was a Postdoctoral Researcher with Yonsei University. Since 2019, he has been a Postdoctoral Researcher with the School of Computer and Communication Sciences, École Polytechnique Féd érale de Lausanne (EPFL), Lausanne, Switzerland. His current research interests include 2D/3D computer vision, computational photography, and machine learning.

Dongbo Min (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea, in 2003, 2005, and 2009, respectively. From 2009 to 2010, he was a Postdoctoral Researcher with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA. From 2010 to 2015, he was with Advanced Digital Sciences Center, Singapore. From 2015 to 2018, he was an Assistant Professor with the Department of Computer Science and Engineering, Chungnam National University, Daejeon, South Korea. Since 2018, he has been an Assistant Professor with the Department of Computer Science and Engineering, Ewha Womans University, Seoul. His current research interests include computer vision, deep learning, video processing, and continuous/discrete optimization.

Kwanghoon Sohn (Senior Member, IEEE) received the B.E. degree in electronic engineering from Yonsei University, Seoul, South Korea, in 1983, the M.S.E.E. degree in electrical engineering from the University of Minnesota, Minneapolis, MN, USA, in 1985, and the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, NC, USA, in 1992. He was a Senior Member of Research Engineering with Satellite Communication Division, Electronics and Telecommunications Research Institute, Daejeon, South Korea, from 1992 to 1993, and a Postdoctoral Fellow with the MRI Center, Medical School of Georgetown University, Washington, DC, USA, in 1994. He was a Visiting Professor with Nanyang Technological University, Singapore, from 2002 to 2003. He is currently an Underwood Distinguished Professor with the School of Electrical and Electronic Engineering, Yonsei University. His research interests include 3D image processing and computer vision.